



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/602,300	06/23/2000	G. Eric Engstrom	3382-55510	1178

7590 06/11/2004

KLARQUIST SPARKMAN CAMPBELL  
LEIGH & WHINSTON, LLP  
One World Trade Center Suite 1600  
121 S W Salmon Street  
Portland, OR 97204-2988

EXAMINER

PATEL, HARESH N

ART UNIT	PAPER NUMBER
2154	9/8

DATE MAILED: 06/11/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Supplemental Notice of Allowability</b>	Application No.	Applicant(s)
	09/602,300	ENGSTROM ET AL.
	Examiner	Art Unit
	Haresh Patel	2154

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1.  This communication is responsive to 3/15/04.
2.  The allowed claim(s) is/are 1-10 and 20-27.
3.  The drawings filed on 23 June 2000 are accepted by the Examiner.
4.  Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
  - a)  All
  - b)  Some\*
  - c)  None
 of the:
  1.  Certified copies of the priority documents have been received.
  2.  Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3.  Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

\* Certified copies not received: \_\_\_\_\_.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.  
**THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.**

5.  A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
6.  CORRECTED DRAWINGS ( as "replacement sheets") must be submitted.
  - (a)  including changes required by the Notice of Draftsperson's Patent Drawing Review ( PTO-948) attached
    - 1)  hereto or 2)  to Paper No./Mail Date \_\_\_\_\_.
  - (b)  including changes required by the attached Examiner's Amendment / Comment or in the Office action of
 Paper No./Mail Date \_\_\_\_\_.
- Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
7.  DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

**Attachment(s)**

1.  Notice of References Cited (PTO-892)
2.  Notice of Draftperson's Patent Drawing Review (PTO-948)
3.  Information Disclosure Statements (PTO-1449 or PTO/SB/08),  
Paper No./Mail Date \_\_\_\_\_
4.  Examiner's Comment Regarding Requirement for Deposit  
of Biological Material
5.  Notice of Informal Patent Application (PTO-152)
6.  Interview Summary (PTO-413),  
Paper No./Mail Date 8.
7.  Examiner's Amendment/Comment
8.  Examiner's Statement of Reasons for Allowance
9.  Other See Continuation Sheet.



JOHN FOLLANSBEE  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100

Continuation of Attachment(s) 9. Other: Applicant's claim for domestic priority under 35 U.S.C. 119(e) is acknowledged. This application benefits the priority date 09/24/1997 of patent number 6,134,602 .

### EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.
2. Authorization for this examiner's amendment was given in a telephone interview with Stephen A Wight on May 14, 2004.
3. The application has been amended as follows:

This listing of claims replaces all prior versions, and listings, of claims in the application.

#### Listing of Claims:

*Subj* 1. (Currently amended) In a multitasking operating system that uses virtual memory to share physical memory among concurrently executing application programs, a method for controlling allocation of physical memory comprising:

*D* in response to a program call from an application program[,] other than an operating system, create a data structure to group said application specified code or data in a group [creating a structure to group the code or data specified by the application];

monitoring for a not-present interrupt generated by a virtual memory system of said multitasking operating system in response to a said application request to access any part of the code or the data in the group; and

when the not-present interrupt occurs for a unit of memory in the group, loading all of the code or the data in the group that is not already in physical memory into said physical memory from secondary storage at one time, using a single series of loading operations without further non-present interrupts being generated by said virtual memory system for another unit of memory in said group, and said loading including [loading] the unit of memory [for which the not present interrupt has occurred] and all other units of memory used to store the code or the data in the group.

Art Unit: 2154

2. (Original) The method of claim 1, wherein the structure includes a linked list structure that links together code or data stored at non-contiguous portions of virtual memory.

3. (Original) The method of claim 2 wherein the structure links pages of memory associated with the non-contiguous portions of code or data.

4. (Original) The method of claim 1, further including:  
repeating the steps of claim 1 for additional groups of code or data specified by the application.

5. (Original) The method of claim 4 further including:  
repeating the steps of claim 1 for a group of code or data for another concurrently executing application such that more than one concurrently executing application program has specified at least one group of code or data to be treated as a single piece of memory for loading into physical memory in response to a not-present interrupt.

6. (Original) The method of claim 1 further including:  
when the not-present interrupt occurs, checking whether the interrupt has occurred for a unit of memory in the group by evaluating whether an address of the memory request for which the interrupt occurred is within a series of non-contiguous memory addresses of the group.

7. (Original) The method of claim 1, further including:  
tracking memory accesses to units of memory in the group together such that when a unit of memory in the group is accessed, all of the units of memory in the group are marked as accessed; and

determining which portions of physical memory to swap from physical memory to secondary storage by determining which units of code are marked as accessed, such that units are selected to be swapped from physical memory to secondary storage based on frequency of use or how recently the units of code have been accessed.

8. (Original) The method of claim 7 further including:  
in response to a call from an application program to group specified code or data in a second group, creating a second structure to group the code or data specified by the application;  
tracking memory accesses to units of memory in the first and second group such that when a unit of memory in both the first and second group is accessed, all of the units of memory in the first and second group are marked as accessed and the unit of memory in both the first and second group is marked as being accessed twice.

9. (Original) The method of claim 8  
when a block of code or data shared between two or more groups is accessed, marking the block as being accessed n times where n is the number of groups that share the block.

10. (Currently amended) A computer-readable medium storing instructions for performing the steps of a method recited in claim 1.

Art Unit: 2154

- D |
11. (Canceled)  
12. (Canceled)  
13. (Canceled)  
14. (Canceled)  
15. (Canceled)  
16. (Canceled)  
17. (Canceled)  
18. (Canceled)  
19. (Canceled)

20. (Currently Amended) A computer-readable medium having stored thereon a data structure used for virtual memory management in a multitasking operating system, comprising:

a series of data fields forming a group for indicating blocks of code or data [associated with] specified by a program call by an application to be treated as a single unit for purposes of virtual memory management, the data fields including a list of memory addresses of the blocks and sizes of each block in the list;

wherein the data structure is evaluated in a data processing operation to load each of the blocks into physical memory whenever a not-present interrupt is generated by said virtual memory system in response to said application request for any memory address referring to a location included in one of the said blocks[.];

wherein the loading of the blocks into the physical memory is performed at one time, using a single series of loading operations without further non-present interrupts being generated by said virtual memory system for another unit of memory in said group.

21. (Previously Presented) The computer readable medium of claim 20, wherein the list of memory addresses is an array of pointers to blocks of memory to be placed in the group.

22. (Previously Presented) The computer readable medium of claim 20, wherein the sizes of each block in the list is indicated in an array of parameters.

23. (Previously Presented) The computer readable medium of claim 20, wherein the data structure is used to derive a linked list structure for keeping track of pages used to store the code or data associated with the group as specified by the application.

24. (Previously Presented) The method of claim 1 further comprising, in response to a second call from the application program to further add units of memory to the group, adding the units of memory to the data structure as specified by the application.

25. (Previously Presented) The method of claim 1 further comprising, in response to a second call from the application to delete specified units of memory from the group, deleting the units of memory specified by the application from the data structure.

26. (Previously Presented) The method of claim 1 further comprising, in response to a second call from the application to destroy the group, destroying the data structure previously used for creating the group.

27. (Currently Amended) In a multitasking operating system that uses virtual memory to share physical memory among concurrently executing application programs, a virtual memory management system comprising:

means for creating a data structure to group code or data specified by one of the concurrently executing applications in a group, in response to a program call from the application other than an operating system [to create a group of specified code or data];

means for monitoring for a not-present interrupt generated by a virtual memory system of said multitasking operating system in response to a said application request to access any part of the code or the data in the group; and

means for when the not-present interrupt occurs for a unit of memory in the group, loading all of the code or the data in the group that is not already in physical memory into said physical memory from secondary storage at one time, using a single series of loading operations without further non-present interrupts being generated by said virtual memory system for another unit of memory in said group, and said loading including [means for loading] the unit of memory [for which the not present interrupt has occurred] and all other units of memory used to store the code or the data in the group[, when the not-present interrupt occurs for a unit of memory in the group].

#### DETAILED ACTION

4. The amendment on May 14, 2004 is noted and made of record.
5. Claims 1-10, 20-27, are presented for examination. Claims 11-19 are cancelled.

6. Claims 1-10, 20-27, are allowed.

7. The following is an examiner's statement of reasons for allowance:

Applicant's invention discloses a method, implementation of a data structure stored on a computer-readable medium and a virtual memory management system for using an application program call (other than an operating system call) to specify and load a code or data into the physical memory.

The cited teaching provides an API that enables applications to classify code and data in a group that is to be loaded into physical memory together whenever an application attempts to access any part of the code or data in the group. This API enables applications to improve performance in virtual memory systems because it prevents page faults from being spread out over time as an application makes read/write requests to code and data. The application specifies code and data that is part of a group. When the virtual memory system in the operating system signals a Not present interrupt due to the application's attempt to access code or data in the group that is not present in physical memory, the API implementation causes all of the code and data in the group to be loaded into physical memory together, i.e., in a single series of memory loading operations before the application resumes execution (generation of second not present interrupt). Thus, the latency of loading code or data from secondary storage is compressed into one period of time, and from then on, all of the code and data in the group is in physical memory.

The prior arts do not teach an API implementation to cause all of the code and data in the group to be loaded into physical memory together, i.e., in a single series of memory loading operations before the application resumes execution. Therefore, the claims are allowable over the art of record.

Art Unit: 2154

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

***Conclusion***

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Haresh Patel whose telephone number is (703) 605-5234. The examiner can normally be reached on Monday, Tuesday, Thursday and Friday from 10:00 am to 8:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Follansbee, can be reached at (703) 305-8498.

The appropriate fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Haresh Patel

June 10, 2004.



JOHN FOLLANSBEE  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100